МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №12

По дисциплине «Основы алгоритмизации и программирования»

За 2 семестр

Тема: «Индексные файлы»

Выполнила:

студентка 1 курса

группы АС-56

Карпенко М.В.

Проверил:

Гирель Т.Н.

Брест 2020

*Цель работы:* Изучить принципы программирования с использованием индексных файлов.

*ЗАДАНИЕ:*В программу разработанную в лабораторной работе 9 внести следующие изменения и дополнения:  
 1. При запуске программы данные читаются в массив структур из файла, при добавлении новой записи в массив структур в файл должна дописываться новая запись, без изменения остальных записей.  
 2. Все изменения (изменения полей записи, удаление записи) – сохраняются в файле при помощи перезаписи содержимого всего файла.   
 3. Сортировка должна выполняться по двум полям на выбор при помощи создания индексных файлов. Содержимое индексного файла переписывается в случае изменения значения ключевого поля (поля, по которому выполняется сортировка) или в случае удаления, добавления записей.

*Код программы:*

#include <iostream>

#include <fstream>

#include<string>  
#include <windows.h>

using namespace std;

struct ORDER

{

string score1;

string score2;

int sum;

};

void in(string path)

{

ofstream file;

file.open(path, ios::app);

string str;

cout << "Введите расчетный счет плательщика : " << endl;

cin >> str;

file << str << "\n";

cout << "Введите расчетный счет получателя: " << endl;

cin >> str;

file << str << "\n";

cout << "Введите перечисляемую сумму в рублях: " << endl;

cin >> str;

file << str << "\n";

file.close();

};

int sToi(string str)

{

int data = 0;

int flag = 1;

for (int i = str.length() - 1; i >= 0; i--)

{

data += ((int)str[i] - 48) \* flag;

flag \*= 10;

}

return data;

}

ORDER formStage(string arr[3])

{

ORDER\* flag = new ORDER();

flag->score1 = arr[0];

flag->score2 = arr[1];

flag->sum = sToi(arr[2]);

return \*flag;

};

void fromFile(string path, ORDER\* data)

{

ifstream file;

file.open(path);

string str;

string arr[3];

int i = 0;

int j = 0;

if (!file.is\_open())

{

cout << "ERROR!!!" << endl;

}

else

{

while (getline(file, str))

{

arr[i] = str;

i++;

if (i == 3)

{

ORDER flag = formStage(arr);

data[j] = flag;

j++;

i = 0;

}

}

}

file.close();

};

int main()

{

setlocale(0, "");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

ifstream file; //для чтения файла

ofstream index; //для записи в файл

//объявление переменных, содержащих путь к файлам

string path = "file.txt";

string ind = "index.txt";

//открытие файлов

file.open(path);

index.open(ind);

if (!file.is\_open() || !index.is\_open()) //проверка на открытие файлов

{

cout << "ERROR!!!" << endl;

}

else

{

int N = 0;

int temp;

cout << "Введите 0, чтоб добавить структуры, 1-пропустить\n" << endl;

cin >> temp;

if (!temp) //

{

file.close();

int k;

cout << "Введите количество структур" << endl;

cin >> k;

for (int i = 0; i < k; i++) //цикл для добавления k структур в файл

in(path);

file.close();

}

ifstream file;

file.open(path);

string str;

while (getline(file, str)) //цикл для подсчета количества структур

N++;

file.close();

N /= 3;

ORDER\* arr = new ORDER[N]; //указатель на динамический массив

fromFile(path, arr);

cout << "0-Сортировка по расчетному счету плательщика \n1-Сортировка по сумме" << endl; //начало выполнения сортировки по двум полям

int n;

cin >> temp; //выбираем по какому полю сортируем

if (temp)

{

for (int k = 0; k < N; k++)

{

for (int i = 0; i < N - 1; i++)

{

if (arr[i].sum > arr[i + 1].sum)

{

ORDER temp;

temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

}

}

}

for (int i = 0; i < N; i++) //запись отсортированного массива во второй файл

{

index << arr[i].score1 << "\n";

index << arr[i].score2 << "\n";

index << arr[i].sum << "\n";

}

}

else

{

for (int k = 0; k < N; k++) //сортировка

{

for (int i = 0; i < N - 1; i++)

{

if (arr[i].score1[0] > arr[i + 1].score1[0])

{

ORDER temp;

temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

}

}

}

for (int i = 0; i < N; i++)

{

index << arr[i].score1 << "\n";

index << arr[i].score2 << "\n";

index << arr[i].sum << "\n";

}

}

}

//закрытие файлов

file.close();

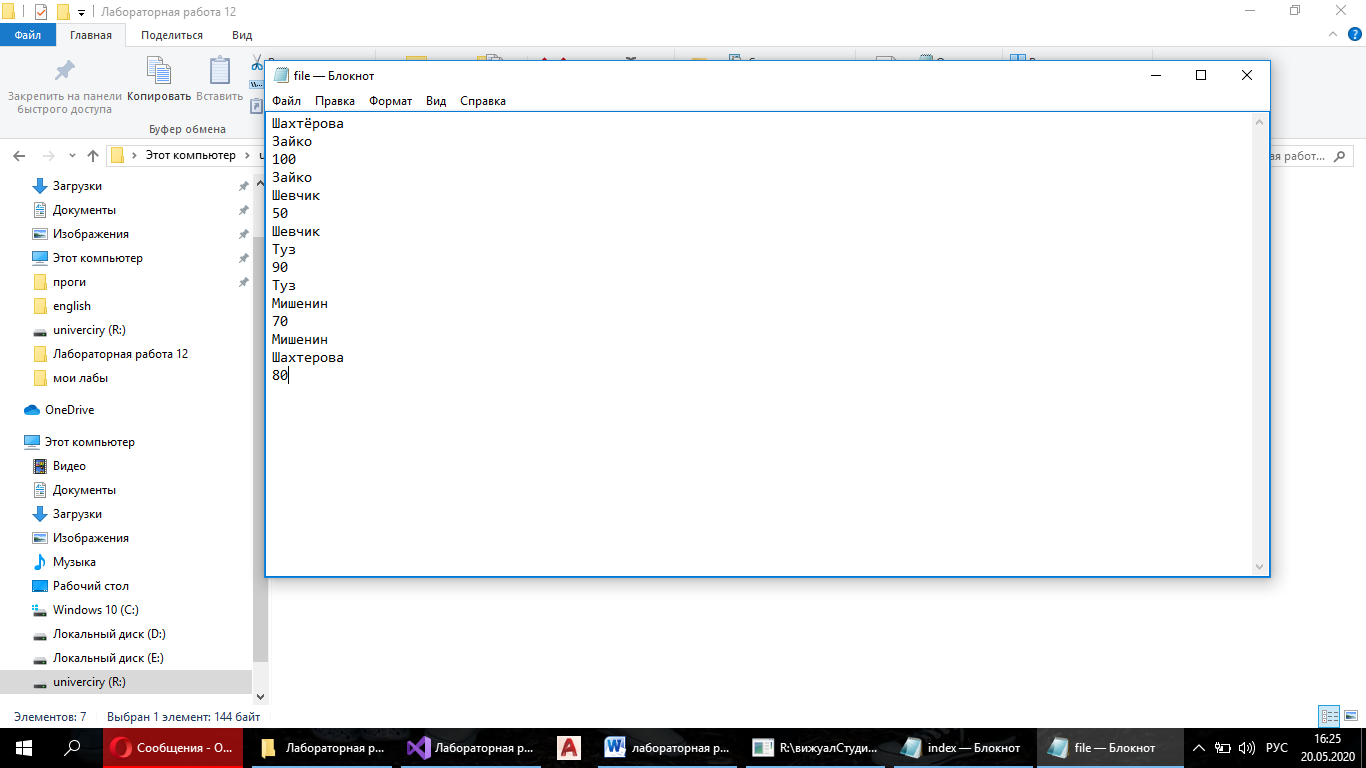
index.close();

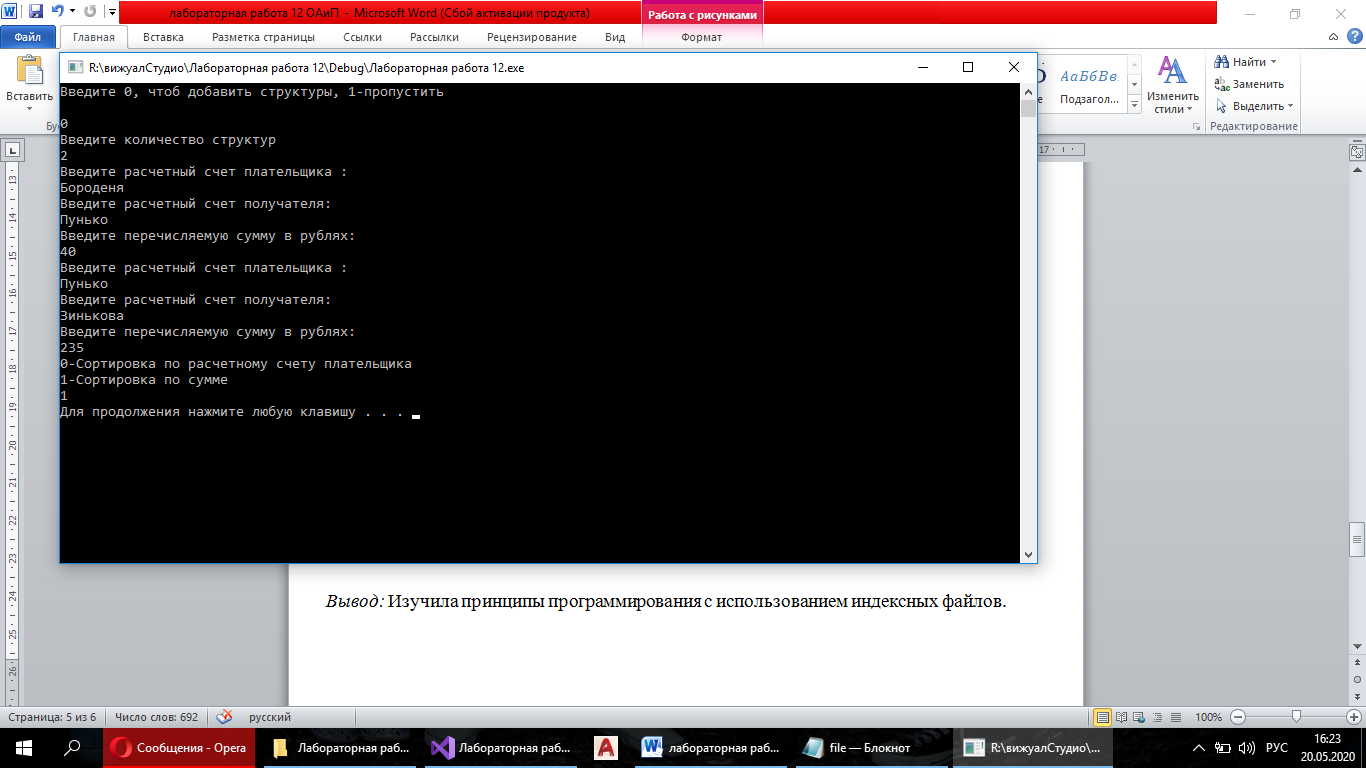
system("pause");

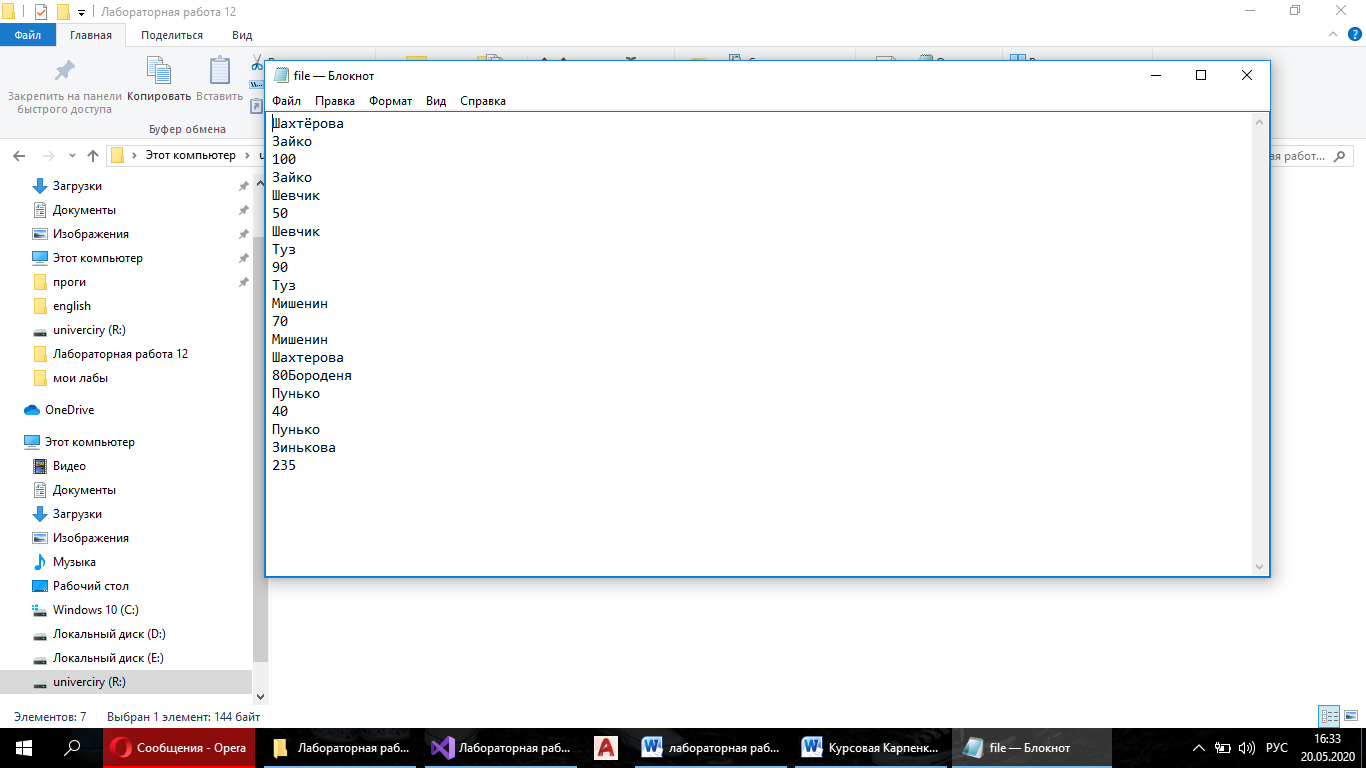
return 0;

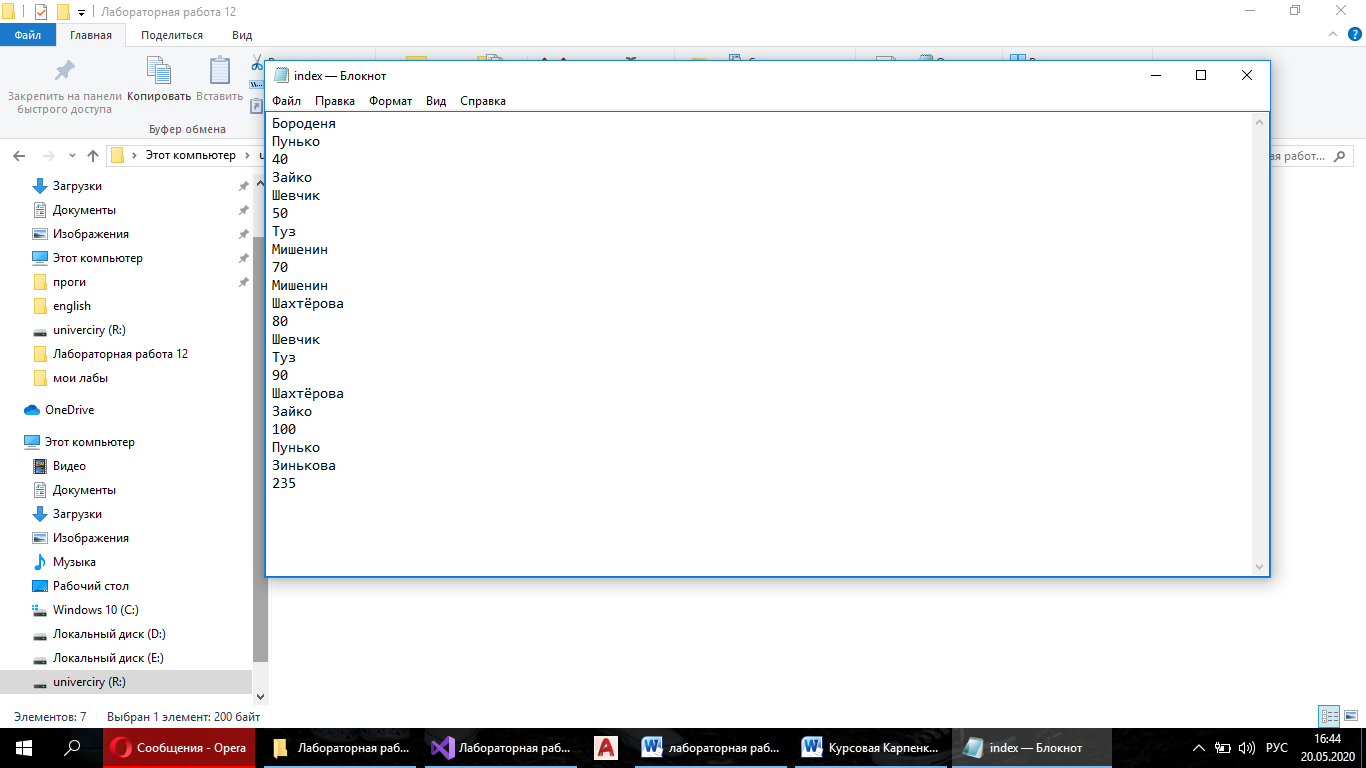
}

*Вывод рез* *ультатов:*









*Вывод:* Изучила принципы программирования с использованием индексных файлов.